고속 다이아몬드 탐색 알고리즘을 위한 MMAD 연산법

서은주[†] · 김동우[†] · 한재혁[††] · 안재형[†††]

요 약

일반 적인 고속 블록정합 알고리즘들은 현재 프레임의 탐색 블록과 참조 프레임의 탐색영역 내의 블록간 MAD(Mean Absolute Distance)를 비교하기 때문에 탐색점 수만큼 MAD를 구해야 하는 단점이 있다. 이와 같은 고속 블록정합 알고리즘들의 단점을 해결하기 위해 "이웃한 화소는 서로 간에 거의 같은 값을 지니고 있다"라는 특성을 이용한 MMAD 연산법에 의한 고속 DS알고리즘을 제안하였다.

본 논문에서는 각 탐색 단계별로 새로운 탐색점의 MAD를 산출하는 기존의 MAD 연산법을 탈피하고, 이전 단계에서 연산된 MAD를 MMAD 연산법에 의해 현재의 탐색점의 MAD로 사용하기 때문에 불필요한 탐색점의 블록간 MAD 연산을 없앨 수 있었다. 움직임 추정 오차는 기존 고속 블록정합 알고리즘과 비교했을 때 비슷한 성능을 나타내었고, 총 연산량은 $2FN^2Ep$만큼 크게 줄어들었다.

MMAD Computation for Fast Diamond-Search Algorithm

Eun-ju Seo[†], Dong-woo Kim[†], Jae-hyeck Han[††] and Jae-hyeong Ahn[†††]

ABSTRACT

Ordinary high-speed block matching algorithms have a disadvantage that they need to get MAD (Mean Absolute Distance) as many as the number of search points due to comparing the MAD between the current frame's search block and the reference frame's search block. To solve such disadvantage of high-speed block matching algorithm, the proposed high-speed DS algorithm employs a MMAD calculation method using a specific characteristic that neighboring pixels have almost same values.

In this thesis, we can get rid of unnecessary MAD calculation between the search point blocks by the new calculation method which uses the previously calculated MAD as the current search point and by breaking from the established MAD calculation method which calculates the MAD of a new search point by each search stage. Comparing with the established high-speed block matching algorithm, this new calculation's estimated movement error was shown as similar, and the total calculation amount decreased by $2FN^2Ep$.

1. 서 론

텍스트나 오디오 정보는 비교적 데이터의 양이 적어 정보의 저장이나 전송에 큰 문제가 없지만 정지영상이나 동영상을 표현하기 위해서는 그 자체의 속성이 다른 신호들과는 달리 정보량이 많다. 그러므로 제한된 용량의 저장매체에 저장하거나 용량이 적은 전송 채널을 통하여 전송하는데 큰 문제점을 가지게 된다. 이러한 영상 정보를 효율적으로 처리하기 위해서는 영상 정보의 압축이절대적으로 필요하다.

동영상 정보 압축의 기본은 인접하는 화면들 사이의 시간적인 중복 정보(Redundant information)를 움직임 추정(Motion estimation)과 움직임 보상(Motion compensation)방법을 사용하여 줄이고,이산 코사인 변환(Discrete cosine transform) 방법을 이용

[†] 준회원, 충북대학교 정보통신공학과 석사과정
[††] 충북대학교 정보통신공과 박사과정
[†††] 정회원, 충북대학교 전기전자공학부 교수

하여 공간적인 중복 정보를 제거하는 것이다. 그 중에서도 움직임 추정 및 보상 기법이 가장 중요하다.

움직임 추정 및 보상 기법은 화소 순환 알고리즘(Pel-recursive algorithm; PRA)[1]과 블록정합 알고리즘(Block matching algorithm; BMA)[1]으로 분류된다. PRA는 연산량이 많아 복잡하고, 수렴하지 않는 문제가 발생하기 때문에 잘 사용되지 않는다. BMA는 수행 시간이 비교적 적게 소요되고, 하드웨어구현이 용이하기 때문에 많이 사용되고 있다. BMA는 블록 단위로 참조 프레임의 탐색영역 내에서 정합 오차가 최소인 위치를 찾아 그 위치를 해당 블록의 움직임 벡터로 결정하는 방식이다. BMA를 이용하여 움직임 추정을 수행하는 경우 일반적으로 가장 좋은 성능을 얻을 수 있는 방법으로 전역 탐색 알고리즘(Full search algorithm ; FSA)이 있다.

FSA는 탐색 영역내의 모든 탐색점에 대해 변위 벡터의 오차를 연산한 후 정합 척도가 최적인 블록을 찾기 때문에 오차는 적지만 연산량이 많은 단점이 있다. 이와 같은 단점을 줄이기 위하여 LOGSA(2-D Logarithmic Search Algorithm)[2], TSSA(Three-Step Search Algorithm)[3], CDSA(Con -jugate Direction Search Algorithm)[4], CSA (Cross Search Algorithm)[5], NTSSA(New Three- Step Search Algorithm)[6], 4SSA(Four-Step Sea -rch Algorithm)[7], BBGDSA(Block-Based Gradie -nt Descent Search Algorithm)[8]등과 같은 다양한 방법의 고속 BMA들이 제안되었다. 이러한 방법들은 움직임 추정 오차가 움직임 방향으로 단조 감소한다는 가정을 이용하여 탐색영역의 일부분에 대해서만 탐색을 수행한다. 그 결과 연산량은 줄일 수 있지만 FSA에 비해 예측 오차가 크고 움직임 벡터의 신뢰성이 떨어지는 문제점을 내포하고 있다. 즉, 이 방법들은 움직임 추정 오차를 감안함으로써 연산량의 감소를 얻는다. 이러한 문제를 해결하기 위해 최근 DSA(Diamond search algori -thm; DSA)[9]가 제안되었다. DSA는 이전 단계에서 연산한 탐색점의 MAD(Mean abso-lute distance; MAD)를 현재의 탐색 단계에서 이용하기 때문에 탐색점 수를 크게 줄일 수 있다. DSA의 움직임 추정 오차는 최소 MAD가 중심값이 될 때까지 탐색을 수행함으로써 개선된다.

본 논문에서는 각 탐색 단계별로 새로운 탐색점의 MAD를 산출하는 기존의 MAD 연산법을 탈피하고, 이전 단계에서 연산된 MAD들의 평균값을 현재의 탐색점으로 사용하는 고속 DSA를 위한 MMAD 연산법을 제안하였다. 제안한 알고리즘에 사용된 연산법은 탐색 최종 단계에서 이용되는 새로운 SDSP (Small Diamond Search Pattern)의 MAD 연산에 적용되므로써 총 연산량을 크게 줄일 수 있었고, DSA 보다 고속으로 움직임 추정이 가능하였다. 실험을 통하여 제안한 알고리즘과 기존 알고리즘을 비교 분석하였고 총 연산량이 크게 줄어들었음을 보였다.

2. 블록정합 알고리즘 기법

BMA는 현재 입력 프레임을 일정한 크기의 블록으로 나눈 후, 나눈 각각의 블록에 대해 참조 프레임에서 그 블록과 가장 유사한 블록을 찾아내어 움직임 벡터를 얻어내는 방법이다.

정합 블록 결정은 참조 프레임에서 설정된 탐색 영역 내에 식(1)을 이용하여 최소 차분 값을 갖는 블록으로 결정한다. 두 블록 사이의 차분 값은 평균 자승 오차(Mean squared difference; MSD)와 비슷한 성능을 유지하면서도 연산 량이 적고 하드웨어 구현이 용이한 MAD가 널리 이용되고 있다.

$$MAD_{(k,l)}(x,y) = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|I_t(k+i,l+i) - I_{t-1}(k+x+i,l+y+j)| \quad (1)$$

식(1)에서, N은 블록의 수직 및 수평 방향 크기, $I_t(i, j)$는 현재 프레임의 (i, j)좌표에서 화소의 휘도 값, $I_{t-1}(i, j)$는 참조 프레임의 (i, j)좌표에서 화소의 휘도 값, (k, l)은 블록의 위치 좌표, 그리고 (x, y)는 탐색 영역에서의 탐색점 위치를 나타낸다. 이를 이용한 (k, l)번째 블록의 움직임 벡터는

$$v(k,l) = \arg\min_{(x,y)} MAD_{(k,l)}(x,y) \quad (2)$$

와 같이 구한다.

BMA가 적용된 움직임 추정의 경우 일반적으로 가장 좋은 성능을 얻기 위해 정합될 수 있는 모든 화소에 대해 MAD를 구한 뒤 그 중 가장 작은 MAD를 갖는 탐색점 (x, y)의 값을 움직임 벡터로 결정하게된다.

3. 다이아몬드 탐색 알고리즘(DSA)

FSA는 참조 프레임의 탐색영역 내에서 정합 될 수 있는 모든 화소에 대해 MAD를 구하고, 그 값을 탐색점으로 정하여 그들 중 가장 작은 MAD를 갖는 탐색점을 움직임 벡터로 결정하는 것이다. 이 방법에서는 모든 탐색점에 대해 탐색하여 정합 척도가 최적인 블록을 찾기 때문에 오차는 적으나 연산량이 많은 단점이 있다. 이러한 단점을 개선하기 위해 다양한 고속 BMA들이 제안되었다. 이 알고리즘들은 "움직임 추정 오차는 움직임 방향으로 단조 감소한다"는 가정하에 일부 탐색점에 대해서만 탐색을 수행하므로 연산량을 크게 줄일 수 있다. 그러나, 이러한 방법들은 연산량은 줄일 수 있지만 FSA에 비해 움직임 추정 오차가 크고 움직임 벡터의 신뢰성이 떨어지는 문제점을 내포하고 있다. 이러한 문제를 해결하기 위하여 최근에 DSA가 제안되었다. DSA은 대부분의 동영상이 영상의 중심부분에서 움직임이 활발함을 이용하여 중앙 집중적 블록정합을 수행한다. 이 알고리즘은 최소 MAD가 중심값이 될 때까지 탐색을 하고, 이전 단계에서 연산한 탐색점의 MAD를 현재의 탐색 단계에서 이용하므로 탐색점수를 줄인다. 따라서 기존 고속 BMA보다 움직임 추정 오차가 적은 고속의 블록정합이 가능하다.

이 알고리즘은 그림 1과 같은 2가지의 탐색패턴을 이용하여 탐색한다. 탐색 시작단계에서 그림 1(a)의 LDSP(Large Diamond Search Pattern)를 이용하고, 탐색 최종 단계에서 그림 1(b)의 SDSP 를 이용하여 움직임 벡터를 추정한다.

(a) LDSP (b) SDSP

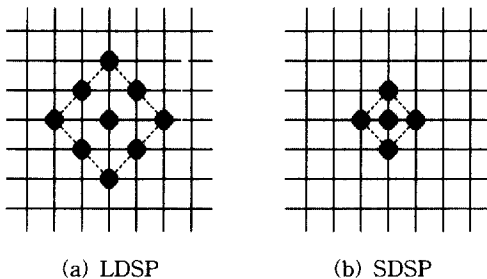



그림 1. DSA의 2가지 탐색점 패턴

탐색 영역을 15×15(-7~+7:수직 및 수평방향 변위)로 두었을 때 탐색 과정은 그림 2와 같다.

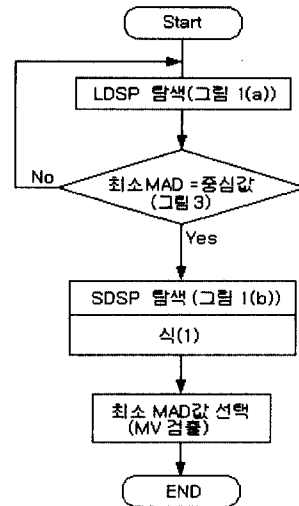



그림 2. DSA의 탐색 과정

그림 4는 DSA가 적용된 움직임 추정 예이다.

DSA는 매 블록 당 탐색영역 내의 각각의 탐색점에 대해 정합 될 수 있는 모든 화소에 대해 MAD를 구하고, 그 중 가장 작은 MAD를 갖는 탐색점의 값을 그림 4와 같이 움직임 벡터로 결정한다.

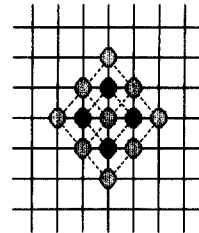



그림 3. 최소 MAD가 중심값일 때

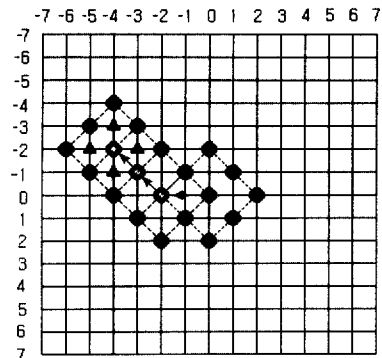



그림 4. DSA의 움직임 추정 예(MV:-2,-4)

4. 제안한 고속 블록정합 알고리즘

기존의 BMA들은 현재 프레임의 탐색 블록과 참조 프레임의 탐색영역 내의 탐색점에서 블록간 MAD를 구하고, 그 값을 탐색점으로 사용하여 탐색점 수만큼 MAD를 구하였다.

본 논문에서는 "이웃한 화소는 서로 간에 거의 같은 값을 지니고 있다"라는 특성을 이용하여, 기존의 MAD 연산법을 탈피하고 고속 DSA를 위한 MMAD 연산법을 제안하였다. 이 연산법은 이전 단계에서 연산된 MAD들의 평균값을 구하여 그 값을 현재 탐색점의 MMAD로 사용한다. 탐색 단계에서 최소 MAD를 갖는 탐색점이 중심값일 때만 식(3)을 적용하고, 그림 5처럼 이웃한 4개의 탐색점의 MAD 평균값을 구하여 그 값을 새로운 탐색점의 MMAD로 사용한다.

$$MMAD = (MAD1 + MAD2 + MAD3 + MAD4) / 4 \tag{3}$$

식(3)과 그림 5의 $MAD1$, $MAD2$, $MAD3$, $MAD4$는 이전 단계에서 구한 탐색점의 MAD이고, $MMAD$는 주변 탐색점들의 MAD 평균값이다.

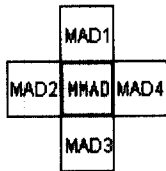



그림 5. MMAD 연산에 사용되는 주변 탐색점들

그림 6은 MMAD 연산법을 적용하여 탐색점의 MMAD를 산출한 예이다.

그림 7은 제안알고리즘에서 쓰인 탐색점 패턴을 나타낸 것이다. 그림 7(a)는 새로운 LDSP를 나타낸

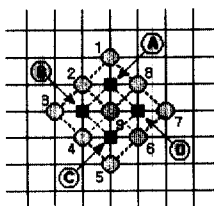



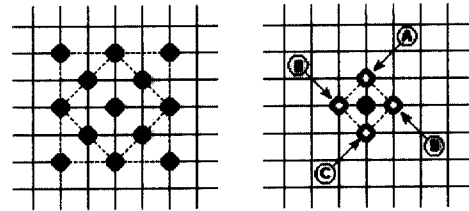
그림 6. 최소 MAD가 중심값일 때 MMAD 연산 예

(a) 새로운 LDSP (b) 새로운 SDSP

그림 7. 제안 알고리즘의 2가지 탐색점 패턴

것이며 DSA의 LDSP 탐색점 수 보다 4개 추가되었다. 제안방법의 새로운 LDSP는 추가된 탐색점 수만큼 블록정합을 수행하므로 DSA보다 정밀한 블록정합을 수행한다.

그림 7(b)의 A,B,C,D는 최소 MAD가 탐색점의 중심값일 때 제안한 MMAD 연산법에 의해 산출된 새로운 SDSP를 나타낸 것이다.

탐색 영역을 15×15(-7~+7:수직 및 수평방향 변위)로 두었을 때, 식(3)의 연산법을 사용한 탐색 과정은 그림 8과 같다.

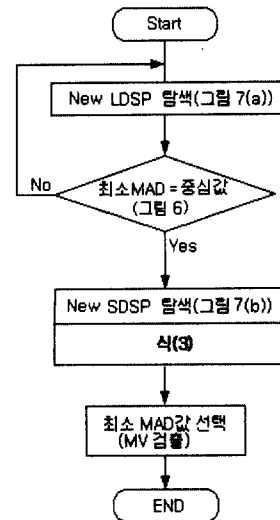



그림 8. 제안 알고리즘 탐색 과정

그림 9는 MMAD 연산법이 적용된 제안 알고리즘의 움직임 추정 예이다.

DSA의 MAD 연산법은 그림 3에서 알 수 있듯이 최소 MAD가 중심값일 때 탐색영역 내에 정합되는 화소들을 식(1)에 적용하여 MAD를 산출한다. 그러

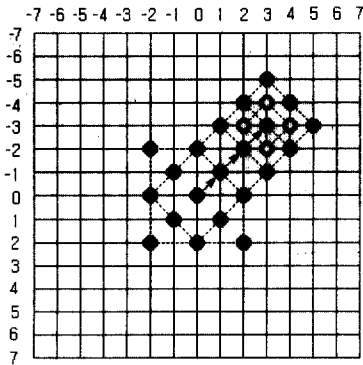



그림 9. 제안한 알고리즘의 움직임 추정 예(MV:3,-3)

나 제안된 알고리즘의 MMAD 연산법은 식(3)을 적용하여 그림 6과 같이 이웃한 탐색점의 평균값에 의해 산출되므로 연산량이 크게 줄어듬을 알 수 있다. 식(4)는 일반 BMA의 총 연산량 산출법을 나타낸 것이고, 식(5)는 제안한 방법의 총 연산량 산출법을 나타낸다.

$$TC = 2FN^2SP_{AVG} \tag{4}$$

$$TC = 2FN^2Sp_{AVG} - (2FN^2Ep) \tag{5}$$

위 식에서 TC는 총 연산량을 나타내며, 상수 2는 MAD연산에서 각1회의 뺄셈과 덧셈연산을 나타내며, F는 한 프레임의 블록 수, N은 블록 크기, SpAVG는 탐색점 수의 평균을 나타내며 Ep는 MMAD 연산법에 의해 구해진 탐색점 수이다.

5. 실험 및 성능 분석

제안한 알고리즘의 성능 평가를 위해 컴퓨터 모의 실험을 수행하였다. 본 실험에서는 그림 10과같이 CIF크기의 gray영상 claire, table tennis, flower garden을 각각 80프레임씩 사용하였다. 움직임 추정에 사용된 블록의 크기는 16x16이며

탐색 영역은 15×15(-7～+7:수직 및 수평방향 변위)를 사용하였다.

표 1은 실험 영상으로 사용된 동영상이 가지는 움직임 특성을 나타낸다.

제안 알고리즘의 성능을 평가하기 위해 FSA, 4SSA, DSA등과 MSE, PSNR비교, 탐색점 수, 총 연산량을 비교하였다.

표 1. 실험 동영상의 움직임 특성

실험 동영상	영상의 움직임 특성
Claire	물체의 느린 이동과 적은 움직임
Table tennis	물체의 빠른 이동과 확대
Flower garden	활동적 움직임과 빠른 파노라마식의 움직임

(a) Claire

(b) Table tennis

(c) Flower garden

그림 10. 실험 동영상들

그림 11은 실험 영상들에 대한 FSA, DSA, 제안한 알고리즘의 MSE를 비교한 것이고, 표 2는 FSA, DSA, 제안 알고리즘의 PSNR을 비교한 것이다.

그림 11과 표 2로부터 제안한 MMAD 연산법이 적용된 제안 알고리즘의 움직임 추정오차가 DSA와 거의 같은 값을 나타냄을 확인할 수 있다.

표 3과 그림 12는 제안한 알고리즘과 4SSA, DSA의 탐색점 수를 비교한 것이다.

(a) Claire

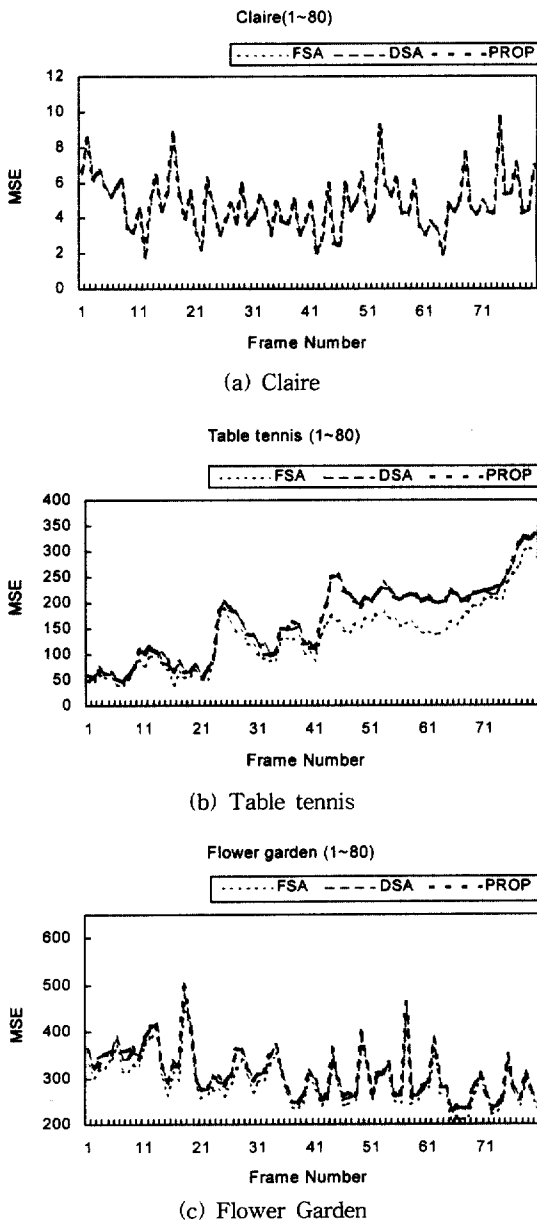
(b) Table tennis

(c) Flower Garden

그림 11. 실험 영상에 대한 MSE비교

표 2. PSNR 비교

BMAs	claire	table tennis	flower garden
FSA	37.85	27.8	26.92
DSA	37.69	27.46	26.8
Prop	[illegible]	[illegible]	[illegible]

표 3. 탐색점 수 비교

BMAs	claire	Table tennis	Flower garden
4SSA	16.2	18.4	19.3
DSA	14.3	16.3	17.1
PROP	[illegible]	[illegible]	[illegible]

이 표로부터 제안방법의 탐색점 수가 DSA에 비해 평균 2개 늘어남을 볼 수 있으며, 이는 새로운 LDSP가 적용되었기 때문이다.

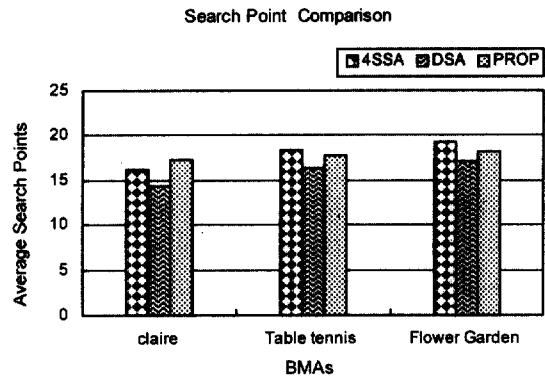



그림 12. 탐색점 수 비교

표 4와 그림 13은 제안한 알고리즘과 4SSA, DSA의 총 연산량을 비교한 것이다.

표 3과 표 4로부터 탐색점 수는 늘었지만 MMAD 연산법을 적용한 제안알고리즘의 연산량이 DSA보다 크게 줄었다는 것을 확인할 수 있다.

표 4. 총 연산량 비교

실험 동영상	기존의 BMAs		제안한 알고리즘	
	Computation		Computation	개선율(%)
	4SSA	DS		
claire	729907.2	644300.8	[illegible]	[illegible]
table tennis	829030.4	734412.8	[illegible]	[illegible]
flower garden	869580.3	770457.6	[illegible]	[illegible]

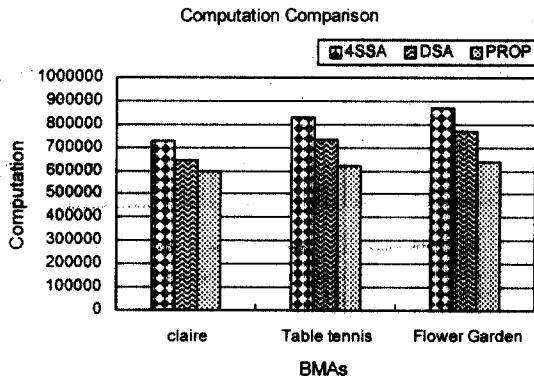



그림 13. 총 연산량 비교

6. 결 론

본 논문에서는 "이웃한 화소는 서로 간에 거의 같은 값을 지니고 있다"라는 특성을 이용한 고속 DSA를 위한 MMAD 연산법을 제안하였다.

제안한 알고리즘은 움직임 추정 오차 측면에서 DSA에 가까운 우수한 움직임 추정 성능을 나타내었다. 또한, 본 논문에서는 연산량을 크게 줄이기 위한 방법으로 각 탐색 단계별로 새로운 탐색점의 MAD를 산출하는 기존의 MAD 연산법을 탈피하고, 이전 단계에서 연산된 MAD를 MMAD 연산법에 의해 현재의 탐색점으로 사용하기 때문에 불필요한 탐색점의 블록간 MAD 연산을 $2FN^2Ep$ 만큼 줄일 수 있었다.

이 방법은 대표적인 고속 블록정합 알고리즘인 4SSA, DSA와 비교하여 총 연산량이 평균 18.3% 개선되었으면서도 뛰어난 움직임 추정 성능을 나타내었다. 제안한 방법의 성능을 평가하기 위한 컴퓨터 모의 실험결과로부터 제안한 방법이 움직임 추정 오차 측면에서 우수한 성능을 유지하면서도 연산량을 현저히 줄일 수 있음을 확인할 수 있었다.

참 고 문 헌

[1] A.Murat,Tekalp, "Digital video processing," pp. 101-106, 117-129, Aug. 1995

[2] J.Jain and A.Jain, "Displacementmeasure-ment and its application in interframe image coding," *IEEE Trans.Commun.*, vol. COMM-29, pp.1799-1808, Dec. 1981.

[3] T.Koga, K. Iinuma, A. Hirano, Y.Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *in Proc. Nat. Telecommun. Conf.*, New Orleans, LA, No.3, pp. G5.3.1-5.3.5. Dec. 1981

[4] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.*, **vol.** COMM-33, pp. 888-896, Aug. 1985.

[5] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, pp. 950-953, July 1990.

[6] R. Li, B. Zeng, and M.L.Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438-442, Aug. 1994.

[7] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 313-317, June 1996.

[8] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 419-423, Aug. 1996

[9] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast blcok matching motion estimation,"*IEEE Trans. Image process.*, vol. 9, pp. 287-290, Feb. 2000.

서 은 주

1997년 2월 충주대학교 컴퓨터공학과(학사)
1997년 9월~1999년 3월 (주) 대하전자 멀티미디어팀 연구원
2000년 3월~현재 충북대학교 정보통신공학과(석사과정)
관심분야 : 영상통신 및 영상정보처리 임베디드 시스템, RTOS

김 동 우

1997년 2월 충북대학교 정보통신공학과(학사)
1997년 3월~2001년 2월 (주)현대전자통신시스템연구소 연구원
2000년 3월~현재 충북대학교 정보통신공학과(석사과정)
관심분야 : 영상처리 및 내용기반검색

한 재 혁

1998년 2월 충북대학교 정보통신공학과(학사)
2000년 2월 충북대학교 정보통신공학과(석사)
2000년 3월~현재 충북대학교 정보보통신공과 (박사과정)
관심분야 : 티미디어 통신, 영상압축

안 재 형

1981년 충북대학교 전기공학과(학사)
1983년 한국과학기술원 전기 및 전자공학과(석사)
1992년 한국과학기술원 전기 및 전자공학과(박사)
1987년~현재 충북대학교 전기 전자공학부 교수
관심분야 : 영상 통신 및 영상정보처리, 멀티미디어 제작 및 정보제공, 인터넷 통신 및 프로그래밍